DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

C448843

WHY STOCHASTIC MODELING IS ESSENTIAL
IN ANALYZING INTERDICTED TRANSPORTATION
NETWORK PERFORMANCE

by

Choi, Byung Kwon

June 1989

Thesis Advisor : Michael P. Bailey

Approved for public release; distribution is unlimited

T244052

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) Code 54	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a NAME OF FUNDING / SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
WHY STOCHASTIC MODELING IS ESSENTIAL IN ANALYZING INTERDICTED TRANSPORTATION NETWORK PERFORMANCE

12 PERSONAL AUTHOR(S)
Choi, Byung Kwon

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ______ TO ______	June 1989	70

16 SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Road Network, Interdiction, Transportation

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

In this thesis, we use simulation to study the probabilistic dynamics of a road transportation network when the network is subjected to interdiction by aircraft. We consider several road interdiction schemes. The simulation results are compared to those obtained by using deterministic expected value optimization techniques. This latter approach has been used by other researchers in studies of interdictions of networks. We demonstrate, through the use of two realistic military examples, that the deterministic results poorly predict the performance of the stochastic systems and that the bias incurred by using deterministic methods is significant. Therefore, the stochastic model should be used in the real situation.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL Michael P. Bailey	22b TELEPHONE (Include Area Code) (408) 646-2085 22c OFFICE SYMBOL 55Ba

DD Form 1473, JUN 86 Previous editions are obsolete SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603 Unclassified

Approved for public release; distribution is unlimited.

Why Stochastic Modeling is Essential in Analyzing Interdicted Transportation Network Performance

Choi, Byung Kwon
Major, Republic of Korea Army
B.S., Republic of Korea Military Academy, 1980

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
June 1989

ABSTRACT

In this thesis, we use simulation to study the probabilistic dynamics of a road transportation network when the network is subjected to interdiction by aircraft. We consider several road interdiction schemes. The simulation results are compared to those obtained by using deterministic expected value optimization techniques. This latter approach has been used by other researchers in studies of interdictions of networks. We demonstrate, through the use of two realistic military examples, that the deterministic results poorly predict the performance of the stochastic systems and that the bias incurred by using deterministic methods is significant. Therefore, the stochastic model should be used in the real situation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I wish to express my gratitude and appreciation to Professor Mike Bailey and Professor Allan W. McMasters for the instruction, guidance and advice throughout this research.

Finally, many thanks to my wife, Dong–Hae and my daughter, Jung–Min, for their love, patience, and support during the past one and half years in Monterey, California.

This paper is dedicated to my daughter Jung–Min.

I. INTRODUCTION

In conventional or limited warfare the success or failure of a military campaign is directly dependent on abilities to transport material some distance overland within the required delivery time. This is particularly true in the case of armed aggression by one nation upon a contiguous or near – contiguous nation, such as might happen in the Korean Peninsula.

The goal of military road network managers is to minimize the time required to transport war materials to their front line troops. The opposing or interdicting forces try to interdict some of the roads to maximize the transit time between the supply source and the front lines by using a variety of means, particularly airstrikes. If the transit time is greater than the required delivery time, the front line troops must either curtail their activity or retreat from the frontal area. Accordingly, it is to the advantage of the interdicting forces to delay the movement of materials as long as possible.

The interdiction forces' target planner knows that the time when a target is hit is as important as which target is hit. In addition, partial destruction of a target using fewer resources may be sufficient for the immediate objective instead of total destruction with an attendant larger number of resources.

Because the application of tactical air interdiction is probably the most effective method of denying the enemy vital war supplies, we will consider airstrikes as our only interdiction means.

The extent to which a target is destroyed is unknown when airstrikes are planned. Aircraft may be shot down by anti–aircraft fire of the road managing forces, they may be prevented from completely destroying the target by defensive gunnery or missiles, or they may fail to completely destroy the target due to human error. Therefore, the level of the success of an attack is probabilistic. As a result of the probabilistic success of attack, repair times, bypass construction times, and transit times after interdiction will also be probabilistic. We will assume that , once the probability of success of attack is known, these three times will be known for certain.

In this thesis, we study the probabilistic dynamics of a road network transportation problem through the use of simulation methodology. We consider several road interdiction schemes and compare these schemes both by using deterministic expected value optimization techniques and by simulation. We will demonstrate, through the use of two realistic military examples, that the deterministic results poorly predict network transit time of the stochastic network and that the bias incurred by using deterministic methods is significant.

Chapter II briefly discusses the assumptions, objective functions, and algorithms of earlier interdiction techniques and models.

Chapter III includes the network description, problem formulation of a deterministic model, the solution procedure for that model and an example.

Chapter IV describes the stochastic network model for which the transit time between source and sink is dependent upon the probabilistic success of an interdiction attack.

Chapter V explains details of the simulation program such as the random number generator and the generation of random transit times.

Chapter VI presents two realistic networks as examples, explains the network data file for each in detail, compares the results from the deterministic and simulation method, and conducts a sensitivity analysis between two models.

Chapter VII presents a summary and conclusions.

II. LITERATURE REVIEW

Current literature in the field of networks contains several different measures of effectiveness for interdicting a military transportation system. These measures include the maximum delay of material transit, the greatest reduction to the maximum flow, and the least accumulated flow over the specified operational period. All of these measures involve single source to single sink, single commodity flows.

Nugent [Ref. 1] presented a method of solving the problem of allocation of effort in the interdiction of a transportation network under the assumption that the damage function was deterministic and exponential. For example, even if a bridge is destroyed, it may still be possible to ford or ferry supplies across a river. The exponential damage function exhibits a decreasing marginal return from interdiction efforts. In addition, he assumed a planar network, divisible effort, single time period, and no storage or depot capability. His objective function was to maximize the reduction in the enemy's resupply capability subject to a constrained number of simultaneous airstrikes. Nugent used the maximal–flow minimal cut theorem and the topological dual to determine which route to select. He did not consider the probability of success of an attack. He used the expected values of flow over a given road segment. As we shall indicate, this use of expected flows detracts from the accuracy of Nugent's results and methods. Finally, he assumed the road managing force would repair the interdicted roads so that they are operational some time after an interdiction.

Sullivan [Ref. 2] developed a method to maximize the time required for an enemy to deliver war material to the front line troops. Any attack was considered completely successful, i.e., the target was fully destroyed. Repair and bypass construction times were assumed to be known linearly decreasing functions of time. He assumed that the resulting transit time over repaired arcs or constructed bypasses was always more than the uninterdicted transit times.

Assuming certain success of all interdicting actions, he was able to solve a deterministic time–dependent shortest route problem using the method of Cooke and Halsey [Ref. 3]. The solution method used by Sullivan was to select a set of arcs to attack based on marginal increases in the source–to–sink delivery delay time.

Mustin [Ref. 4] assumed that upper and lower limits on road capacities, as well as the amount of reduction per sortie, were known deterministically, and the reduction in capacity per sortie was linear between the upper and lower limits of capacity with the latter being positive. The goal of interdicting forces was to prevent the build–up of supplies over a 24–hour period. A computational procedure was developed for determining the optimum strike plan for minimizing network flow capacity. The work of Mustin, Sullivan, and Nugent were all masters' theses at the Naval Postgraduate School under the direction of Alan McMasters.

Wollmer [Ref. 5] presented two algorithms for targeting strikes in a lines–of–communication (LOC) network. He assumed that the user of the LOCs is attempting to achieve a circulation flow at minimum cost, a very general goal that includes, as special cases, maximizing flow between two points, meeting required flows between two points at minimum cost, and combinations of these two. One possibility is to assume all strikes allocated against a particular arc are directed against the same target and each has an identical and independent probability of successfully destroying it. The expected result of the attack is then used in an optimization scheme to maximize the minimum cost of communication.

His algorithms attempt to make such arc costs as large as possible over time while decreasing arc capacities. The first algorithm considers arc costs as linear functions of flow; the second considers arc costs as piecewise linear functions of flow with one break point. Allocation is done on the basis of immediate user cost, repair times, and repair cost. Specifically, if a single strike is to be targeted, it is directed against an arc of maximum strike value, where strike value is defined as the repair cost plus the resulting cost increase per unit time of a minimum–cost circulation flow multiplied by the repair time. For multiple strikes, no method for allocating strikes optimally, other than complete enumeration, is known. Wollmer solved the problem heuristically by repeated application of the one–strike algorithm.

In Sengoku [Ref. 6], two new indices measuring the degree of influence of the arc on the vulnerability of the network were studied. He also assumed the interdiction of each arc was deterministically successful.

The Network Interdiction Model (NIM) [Ref. 7] is a strategic military decision support system which treats three criteria; the maximum delay the interdicted targets would cause, the greatest reduction to the maximum flow, and the least accumulated flow over the specified operational period.

Most of the approaches discussed above were concerned with "deterministic expected values," that is, the expected level of success of an attack was designated deterministically. Some of them assumed that the targets are struck one at a time sequentially. This thesis will examine a stochastic system which incorporates the probability of success of an attack and simultaneous attacks. In the next chapter, the algorithm for determining the time–dependent lengths of arcs will be developed. This algorithm will be used later in both the deterministic and stochastic network models.

III. PROBLEM FORMULATION

A. NETWORK DESCRIPTION

The transportation system can be represented by a network of arcs and nodes. Nodes represent interactions of road segments. Further, they may be used to represent any point at which it is convenient to distinguish between the road characteristics on either side of the node. Arcs represent road segments. Each arc joins two nodes; that is, each is characterized by a beginning node and an ending node. The network is assumed to have undirected capacitated arcs.

Two special nodes are the source and sink. The network is assumed to have a single source through which flow enters and a single sink through which flow leaves. If there are several sources or sinks, this may be taken care of by adding artificial nodes and arcs. The capacities of these artificial arcs are infinite and they are not vulnerable to attack.

We are interested in the transportation times between the source and the sink after several arcs in the network have been attacked and destroyed.

We will follow the network descriptions given in the Sullivan thesis[Ref. 2]. The notation (i, j) represents the arc between nodes i and j. Nodes are numbered from 1 to n with 1 for the source and n for the sink. The intermediate nodes have any of the values between 1 and n. There exists a transit time, d_{ij}, for each arc based upon road conditions, terrain, and time of day. Engineering units are stationed at several prespecified nodes in the network. These units must first travel to an attacked arc, and then perform the repair or make a bypass. Thus, each arc will have attributes of time to move between the closest engineering unit location and the damaged arc, time to repair, or time to construct a bypass. Let

m_{ij} = time to move between the engineering unit location and the damaged arc (i, j),

r_{ij} = time to repair the destroyed structure in arc (i, j),

b_{ij} = time to construct a bypass in the arc (i, j),

$c_{ij} = m_{ij} + r_{ij}$ [or, b_{ij}], time to recover the destroyed arc (i, j),

$c_{ij}(t)$ = time remaining to recover arc (i, j) at time t,

$d_{ij}(r)$ = transit time over arc (i, j) once it has been repaired,

$d_{ij}(b)$ = transit time over a newly constructed bypass of arc (i, j)

The values of $d_{ij}(r)$ and $d_{ij}(b)$ are assumed to be greater than or equal to the uninterdicted transit times, d_{ij}. This is a reasonable assumption since bypass construction is usually inferior to the original road segment, thus slowing traffic. Furthermore, even with a segment fully repaired, a vehicle may have a tendency to travel at a slower pace in an area that has recently undergone a bombing attack.

The transit time can vary with types of military column formations. A fundamental for convoy command and control is that the column be organized to meet the mission requirements and provide the degree of control necessary. The convoy commander decides how his column will be organized for control, choosing from three basic methods: Close Column, Open Column, Infiltration.

The difference in formation depends largely on vehicle spacing. The number of vehicles (density) per kilometer of road and the rate of march are accepted numbers (values) for average conditions when a movement is not influenced by attacking forces' actions. However, weather, tactical situation, attacking forces' capability, condition and type of road, vehicular maintenance, types of vehicles, and command policies may cause changes in average densities and speeds.[Ref. 8: p. 5–4] This thesis will assume the open column which is applicable for a normal supply support situation.

Although any segment of a road is subject to attack, a highway segment that, after attack by strike aircraft, will be extremely difficult to repair or bypass is called a *choke point.* In other words, choke points are segments in a network which once attacked, force an enemy to either reroute traffic or expend large amounts of resources to keep the attacked segment open to traffic. Typical choke points would be bridges, mountain roads, and tunnels. In case no choke points exist in an arc, normal roads are selected for the targets. Each arc is assumed to have a unique attractive target for interdiction. Bypass and recovery times are based on the extent of the destruction of this target.

Immediately following interdiction of a road segment, the arc transit time value becomes a function of recovery time. Therefore, after interdiction, the transit time over arc (i, j) will be defined by:

$$d_{ij}(t) = \min\ [c_{ij}(t) + d_{ij}(r),\ \ c_{ij}(t) + d_{ij}(b)]$$

where t represents the earliest clocktime after attack that a vehicle can depart from the source and arrive at the point of interdiction. The function which is minimized guides the road network manager in deciding how to get the arc back into operation. If $c_{ij}(t) + d_{ij}(r) < c_{ij}(t) + d_{ij}(b)$, then the road manager will choose to repair the road segment. Conversely, if $c_{ij}(t) + d_{ij}(r) > c_{ij}(t) + d_{ij}(b)$, then the manager will choose to construct a bypass around the interdicted point.

An attack is not always successful. An attack may have no effect, it may leave the road damaged but passable, it may cause the road to be impassable with minor repairs, or it may completely destroy the target, necessitating lengthy repairs to restore the road's usefulness. We will call these levels of attack success *no destruction, ruts, partial, and complete,* respectively. Accordingly, the recovery time and the transit time after recovery will be different.

B. DETERMINISTIC PROBLEM FORMULATION

Road network managers want to minimize the time required to transport war material to his front line troops while attacking forces try to maximize the time between source and sink by interdicting this supply operation with airstrikes.

If we assume that the success of each airstrike is known, we could solve a deterministic optimization problem which would give us the length of the required transit time. One method for solving this problem is linear programming, a well known method for performing optimization on networks. However, there exists a special purpose algorithm for this particular optimization problem due to Cooke and Halsey [Ref. 3]. This method was used by Sullivan [Ref. 2].

C. SOLUTION TECHNIQUE

To determine the shortest transit time for a given set of points of interdiction, the Cooke and Halsey algorithm compares the repair and construction bypass functions for each damaged segment along the route. The "length" of interdicted arc is then replaced with $d_{ij}(t)$. The arcs are assumed to be attacked simultaneously within available air sorties.

The algorithm is given in the following steps.

1. Choose the arc sets to attack.

2. The transit time of the arc to be interdicted is replaced by the time dependent length function

$$d_{ij}(t) = c_{ij}(t) + d_{ij}(r),$$

if repair is optimal; otherwise,

$$d_{ij}(t) = c_{ij}(t) + d_{ij}(b).$$

The problem is now time dependent in nature.

3. Define the new tentative node label f_i to be an upper bound on the earliest time of arrival at node i, and permanent label, $f_i{}^p$, to be the earliest possible (optimal) time of arrival.

4. Permanently label node 1 (source) with a value of $f_i = 0$ and label all other nodes with values of infinity; i. e.,

$$f_1{}^{(0)} = 0^p, \; f_2{}^{(0)} = \ldots = f_n{}^{(0)} = \infty.$$

5. Tentatively label all nodes j with the minimum of the current node label f_j and the sum of f_1 and $d_{1j}(t)$; i.e.,

$$f_j{}^{(1)} = \min [\, f_1 + d_{1j}(f_1{}^p), \; f_j{}^{(0)} \,].$$

6. Find the minimum tentative node label; i.e., f_k and declare it permanent.

$$f_k{}^p = \min_{j \neq 1} f_j{}^{(1)}.$$

7. Node k, the new permanent node, is then used to attempt to reduce the labels at all tentatively labeled nodes by comparing $f_k{}^p + d_{kj}(f_k{}^p)$ to the current label. The minimum new temporary label is declared permanent and used as a basis for the next iteration; i.e.,

$$f_j{}^{(m)} = \min [\, f_k{}^p + d_{kj}(f_k{}^p), \; f_j{}^{(m-1)} \,];$$

$$f_k{}^p = \min_{j \neq k} f_j{}^{(m)}.$$

8 Terminate when node n is permanently labeled.

If the network is composed of N arcs and s arcs can be interdicted, there exist C_s^N possible interdiction sets. The interdiction set which has the maximum value of $f_n^{\,p}$ contains the optimal points of attack for the transportation network so as to maximize the time required for the road manager to transport war material to his front line troops. This is the optimal solution to the dual of the road manager's problem.

D. EXAMPLE

Consider the simplified uninterdicted transportation network described by Figure 1. All transit times (d_{ij}) are in minutes, and nodes 1 and 6 will be the source and sink nodes, respectively.

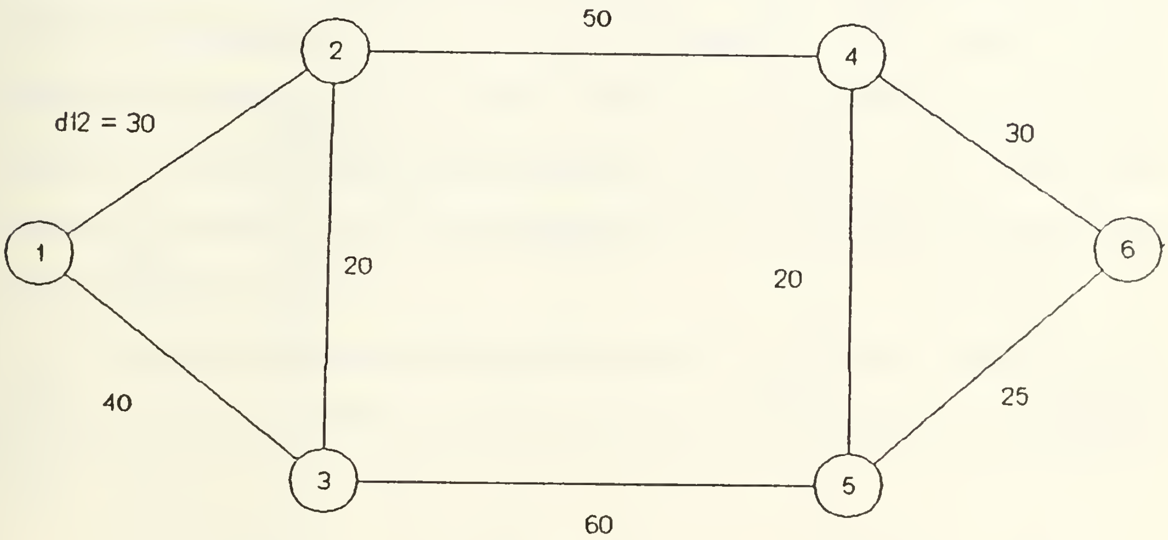

Figure 1. An Example Network.

We believe that the most realistic planning model involves the assignment of a number of aircraft sorties to different network arcs so as to perform a single, simultaneous airstrike. Thus, for our example it is assumed that exactly two arcs can be attacked simultaneously at time 0 with allocated air sorties.

1. The arc set, (2, 4) and (3, 5), will be interdicted.

2. $f_1^{(0)} = 0$, $f_2^{(1)} = 30$, $f_3^{(2)} = 40$.

3. The arc to be interdicted is replaced by the time function. Figure 2 shows the results.

Suppose that arc (2, 4) has the following data:

$m_{24} = 20$,

$r_{24} = 180$, $d_{24}(r) = 60$,

$b_{24} = 200$, $d_{24}(b) = 70$.

Next, $m_{24} + r_{24} + d_{24}(r) = 260 < m_{24} + b_{24} + d_{24}(b) = 290$.

So, $c_{24} = m_{24} + r_{24} = 200$. (the road manager will choose repair.)

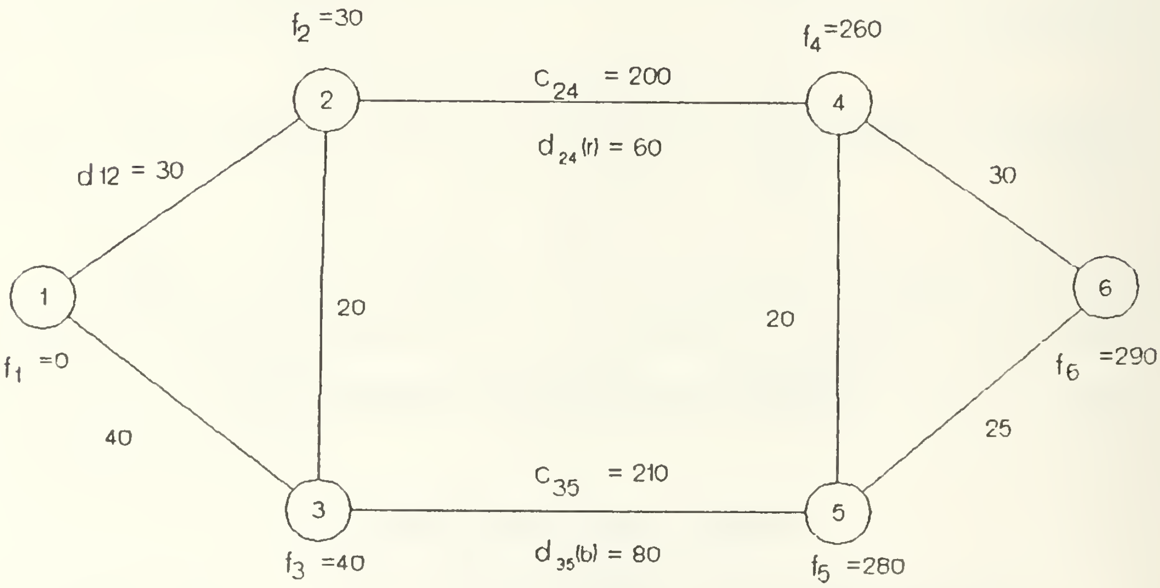

Figure 2. Time–Dependent Network.

According to the time–dependent function,

$$c_{24}(f_2) = 170, \;\; d_{24}(f_2) = c_{24}(f_2) + d_{24}(r) = 230.$$

Next, suppose the data for arc (3, 5) is

$$m_{35} = 30,$$

$$r_{35} = 200, \;\; d_{35}(r) = 70,$$

$$b_{35} = 180, \;\; d_{35}(b) = 80.$$

Then, $m_{35} + r_{35} + d_{35}(r) = 300 > m_{35} + b_{35} + d_{35}(b) = 290.$

So, $c_{35} = m_{35} + b_{35} = 210.$

$$c_{35}(f_3) = 170, \;\; d_{35}(f_3) = 250.$$

4. By using the above results, $f_4^{(3)} = 260$, $f_5^{(4)} = 280$, $f_6^{(5)} = 290$.

The road manager would send his supplies over the shortest route after interdiction. For this example that route is 1 – 2 – 4 – 6. Before sending these supplies he must repair arc (2, 4). He can send supplies out from node 1 at time zero but the trucks will have to wait at node 2 for 170 time units until the arc is repaired. The other alternative is to wait at node 1 for 170 time units before departing to node 2. If this tactic is used then the arc (2, 4) will just be repaired as the trucks arrive at node 2.

In the next chapter, we will consider a model in which the success of a given attack is random, and observe the effects of these attacks on source–to–sink transit time.

IV. STOCHASTIC NETWORK MODEL

The assumption that any attack is considered completely successful is far from the real situation. The target damage may be complete, partial, or only ruts in the road may result from a successful weapon delivery. Also aircraft may be shot down by ground anti–aircraft gunfire or by defensive air patrol interceptors of the road managing forces. Probabilities of kill on the target vary with the levels of pilots' skill, the opponent's defensive gunfire, the exposure time of an aircraft, the type of munition, hardness of target, and so on. Therefore, the level of the success of an attack must be modeled as probabilistic or random instead of assuming that we always have completely successful destruction of the target.

As a result of random successes, repair times, bypass construction time, and transit times after interdiction will also be random variables. Repair times and bypass times will be designated according to the probability of success of attack, i.e., the level of damage on the target. In case of no damage or ruts only, the construction time will be 0. The values after complete destruction are assumed to be greater than or equal to the values after partial destruction. Also, the transit time after interdiction will be influenced by the success of the attack. Even though the interdicted arc is recovered fully, a convoy column commander may have a tendency to travel at a slower pace in an area that has recently undergone a devastating bombing attack because of the psychological effects of being demonstrably vulnerable. It is assumed that the transit times associated with the rut case are larger than the other transit times. This is a reasonable assumption since the road manager will probably not repair a roadway having only minor destruction.

To reiterate, the transit and recovery times of the attacked arcs are random variables which are a function of the effectiveness of the attack. Once the attack's success is known, the transit time and the reconstruction time will be known with certainty.

When faced with random network arc performance, most analysts decide to take the expected value of the arc performance and use this expected value as if it were a deterministic quantity. The expected value is thus input into a deterministic network optimization scheme such as the Cooke and Halsey algorithm above, and the result is reported as the expected value solution. Several of the papers cited in the literature review, particularly Wollmer [Ref. 5], have such published results.

Using expected values in cases such as maximum flow or shortest path problems can be shown to give extremely inaccurate results. To show this we will solve a network maximal flow problem using the expected values and the simulated results. The details are presented in Appendix A. For the simulations we used coins and dice to generate random capacities and calculated the resulting random maximum flows. We also calculated the expected value of the capacity of each arc's capacity and computed the maximum flow on the network using those expected capacities.

As we can see from Table 3 of Appendix A, the result of using the expected capacities is to produce incorrect solutions. The amount of flow calculated using the expected values is 10 and is greater than the result of the average of 6.27 from the stochastic simulation after 30 experiments or trials by a factor of 37.3 per cent.

Using expected values in place of random variables would lead the transportation manager to overestimate his ability to transport goods.

At this point, we can predict similar results for the transit time. We expect that the transportation forces will be able to transport goods to the front much faster than we would be lead to believe had we used the expected time of transit for each arc. This result implies that the algorithms used in all of the deterministic models described in our literature search may not be as efficient as reported. The goal of the remainder of this thesis is to demonstrate this for the Cooke and Halsey algorithm. Chapter VI will demonstrate that the expected transit times are bigger than values calculated using the simulation program we will describe in Chapter V.

V. DISCUSSION OF SIMULATION PROGRAM

If the relationships which comprise the model are simple enough, it may be possible to use analytical methods (such as algebra, calculus or probability theory) to obtain exact information on questions of interest. However, most real–world systems are too complex to allow the models to be evaluated analytically, and these models must be studied by means of computer simulation. In a simulation we use a computer to evaluate a model numerically over a time period of interest, and data are gathered to estimate the desired true characteristics of the model. [Ref. 9: p.1]

A simulation model is said to be deterministic if it contains no random variables. For a deterministic model, there is a unique set of model output data for a given set of inputs. On the other hand, a simulation model is stochastic if it contains one or more random variables. The output data for a stochastic model are themselves random and thus only estimates of the true characteristics of the model. [Ref. 9: p.3]. Our model will be stochastic because the attack successes are random. The output data we will collect are the transit times from the source to the sink, thus our output is indeed random. We will compute statistics based on this data such as the sample mean and sample variance. Our program allows us to specify the number of statistics we desire.

A. PROGRAM OVERVIEW

We wrote our simulation program in Professional FORTRAN, by Ryan McFarland. The program was implemented on an IBM PC compatible computer. Our program begins by inputting the network data file which contains all of the relevant information for the problem. We then enter a loop in which the pseudorandom attack successes are calculated. The resulting deterministic time–dependent shortest path problem is generated and solved using the Cooke and Halsey algorithm, and the resulting shortest path length is recorded.

This loop is repeated the specified number of times. From the recorded statistics, the sample mean and sample variance are calculated and an output file is built which contains the problem, iteration results, and the statistical results. Appendix B contains a listing of the simulation program. We now provide some more detail concerning each stage of the program.

B. NETWORK DATA FILE

As we can see in the Appendix C, the network data file (Table 6) includes the following information concerning the network itself: source, sink, number of nodes, number of arcs, the tail (T) and head (H) node of each arc. For each set of attacked arcs, we specify the recovery times for each type of destruction for each arc (c(1), c(2), c(3), c(4)), the transit time before destruction (d(0)), the transit times after recovery according to the success of attack (d(1), d(2), d(3), d(4)), the arc set attacked, and the probability of each type of destruction is given for each arc (p(1), p(2), p(3), p(4)). Thus, for each attacked set, we have the option of changing any relevant recovery or transit time data.

C. RANDOM NUMBER GENERATOR

Because the success of attack is probabilistic, we utilize pseudorandom numbers, which is conceptually equivalent to flipping coins or rolling dice. This stochastic model will use the linear congruential method (**LCM**) which is one of the most effective and popular types of random number generators. The FORTRAN function RAND() of the program is for generating random numbers.

LAST is the seed which is a locally saved variable. Its value is updated throughout execution of the subroutine that uses RAND. LAST has the value that depends on its previous value and the formula involving LAST, C, and xM. The value of LAST will be between 0 and xM–1 (inclusive) and will be suitably random for proper choices of the constant values xM, C and the starting value of LAST. The values of 29, 217, and 1024 for LAST, C, and xM, respectively, as in RAND, define a usable LCM random number generator. [Ref. 11: pp. 175–178]

With these values of LAST, C, and xM, the function RAND will generate pseudorandom numbers which are approximately uniform on the interval [0, 1]. Because of our relatively small sample sizes, cycle length was not an issue. As an experiment, we also used the recommended generator in Press, Flannery, Teukolsky, and Ve Herling [Ref. 11]. Our results did not differ significantly using these values for LAST, C, and xM.

D. GENERATION OF A RANDOM PROBLEM

We will use different values for different arcs because each arc has different distance, defensive posture, hardness of target, and so on. From these known probabilities we can then construct the associated random number intervals. This process is summarized in Table 1. Suppose that the random number 0.47 is generated, then Table 1 would give the result that the arc was partially damaged.

TABLE 1. RANDOM NUMBER INTERVALS

Level of Damage	Probability (each level)	Cumulative Probability	Interval of random number
No Damage	0.20	0.20	0.00 – 0.20
Ruts Only	0.20	0.40	0.21 – 0.40
Partial	0.50	0.90	0.41 – 0.90
Complete	0.10	1.00	0.91 – 1.00

If the arc sets to attack are decided upon, the probability of success of attack will be designated from using the random number generator and Table 1. Accordingly, the data file then provides the recovery time and the transit time after recovery. Finally, by using the Cooke and Halsey algorithm, we can get the transit time from source to sink.

E. STATISTICAL PROCEDURES

At the end of each series of simulation runs, the subroutine STATS calculates the sample mean and sample variance as well as the 90% and 95% normal confidence intervals for the source–sink transit time. Note that this implies that the simulation model should not be used with less than 30 iterations if one truly desires normal confidence interval results. We determined that, with 200 samples, most of our confidence intervals for the mean transportation time were reasonably tight. Although increasing sample size would have tightened these intervals further, we chose to keep the sample size the same for all cases in the interest of uniformity.

VI. TWO EXAMPLES

In this chapter, we will compare the deterministic model and the stochastic model for two realistic situations: the road network between Kaeseong and Sariwon in North Korea (Example 1) and the road network between Gilroy and Carmel Valley Village in Central California (Example 2). We made reasonable assumptions about moving times, bypass and repair times, and transit times as input into the simulation program. This input we call the network data file. We will first explain the procedure of constructing the network data file. We will then compare the outputs of two models, analyze the results, and interpret what the differences between two models represent.

A. FEATURES COMMON TO BOTH EXAMPLES

It is assumed that the probability of each level of destruction is known and is the same for each arc. The values we selected were shown in Table 1 of the last chapter.

It is assumed that the size of the operational unit in both examples is one North Korean infantry division. The theater–level average consumption rates in a temperate zone are given as 123.7 pounds per person per day [Ref. 12: p.2–5]. If we assume that the number of personnel of the North Korean infantry division is 10,000, the amount of cargo needed to support such a division would be 1,237,000 pounds per day.

Let's assume that the North Korean Army uses trucks which have the same payload and cubic capacity as the South Korean cargo truck, M35A2, 2 1/2–ton.

Its payload is 10,000 pounds on a paved highway and 5,000 pounds offroad [Ref. 8: p.1–7]. We will consider only the highway case in this example. The number of trucks required to move 1,237,000 pounds is 128.

We also assume that the transit time once an arc has been repaired and the transit time over a newly constructed bypass are equal, thus $d_{ij}(b) = d_{ij}(r)$ for each arc (i, j). Although the level of damage, hence the recovery time will be different for each arc, we assumed damage levels which were the same for each arc in our examples. In any operational use of this methodology, we must account for varying levels of damage. Table 2 shows the common degree of damage and recovery time in case of full destruction for each type of target.

TABLE 2. DEGREE OF DAMAGE AND REQUIRED TIME

Type	Degree of Damage	Required Time
Bridge	One pier and two spans	Panel Bridge : 5 hr Bypass Construction : 3 hr
Tunnel	Demolition 50 m from the entrance	Repair : 20 hr Bypass : impossible
Mountainous Road	Demolition 100 m	Repair : 6 hr Bypass : 7 hr
Road	Demolition 150 m	Repair : 5 hr Bypass : 4 hr

Network data files, Table 6 in Appendix C and Table 11 in Appendix D, include the multipliers for inflating the transit and recovery times. For example, the recovery time will be assumed to be zero ($c(1) = c(2) = 0$) in case of no damage or ruts only because the arc is used without repair or bypass. We assume that the recovery time for partial destruction ($c(3)$) is 60 % of the recovery time for complete destruction ($c(4)$). We can get $c(4)$ from Table 4 in Appendix C and Table 9 in

Appendix D. As for transit time, we get d(0) from Table 5 in Appendix C and Table 10 in Appendix D which is the transit time before interdiction (how these times were determined will be described below). Though there is no damage, the transit time will be increased due to psychological effects. In case of ruts only, we can expect considerable delay because road manager forces do not repair or construct a bypass. Furthermore, even with a segment fully repaired, a vehicle may have a tendency to travel at a slower pace in an area that has recently undergone a bombing attack. Consequently, we consider that d(1), d(2), d(3), and d(4) are increased by certain multipliers. (Refer to the end of Table 6 in Appendix C and Table 11 in Appendix D for details)

B. EXAMPLE 1 (KOREA)

Current operational planning for the defense of South Korea is focused on the possibility of a North Korean surprise attack. It is estimated that North Korea may maintain assault pressure with the prepared amounts of supplies in the frontal areas for three days. On the fourth day, additional supplies would be needed for the continued assault. Let us assume that it is the fourth day of the assault. We will consider the network between Kaeseong and Sariwon to be as shown Figure 4. As we can see in Figure 4, the actual road network is composed of 26 nodes and 38 arcs. The source node is Sariwon and the sink node is Kaeseong. Kaeseong is near to the Demilitarized Zone (DMZ) and it is reasonably assumed that the North Korean logistics command is located in Sariwon. This area is mainly composed of mountains, rivers, and rice paddy fields. The road network is well developed in contrast to other areas of North Korea.

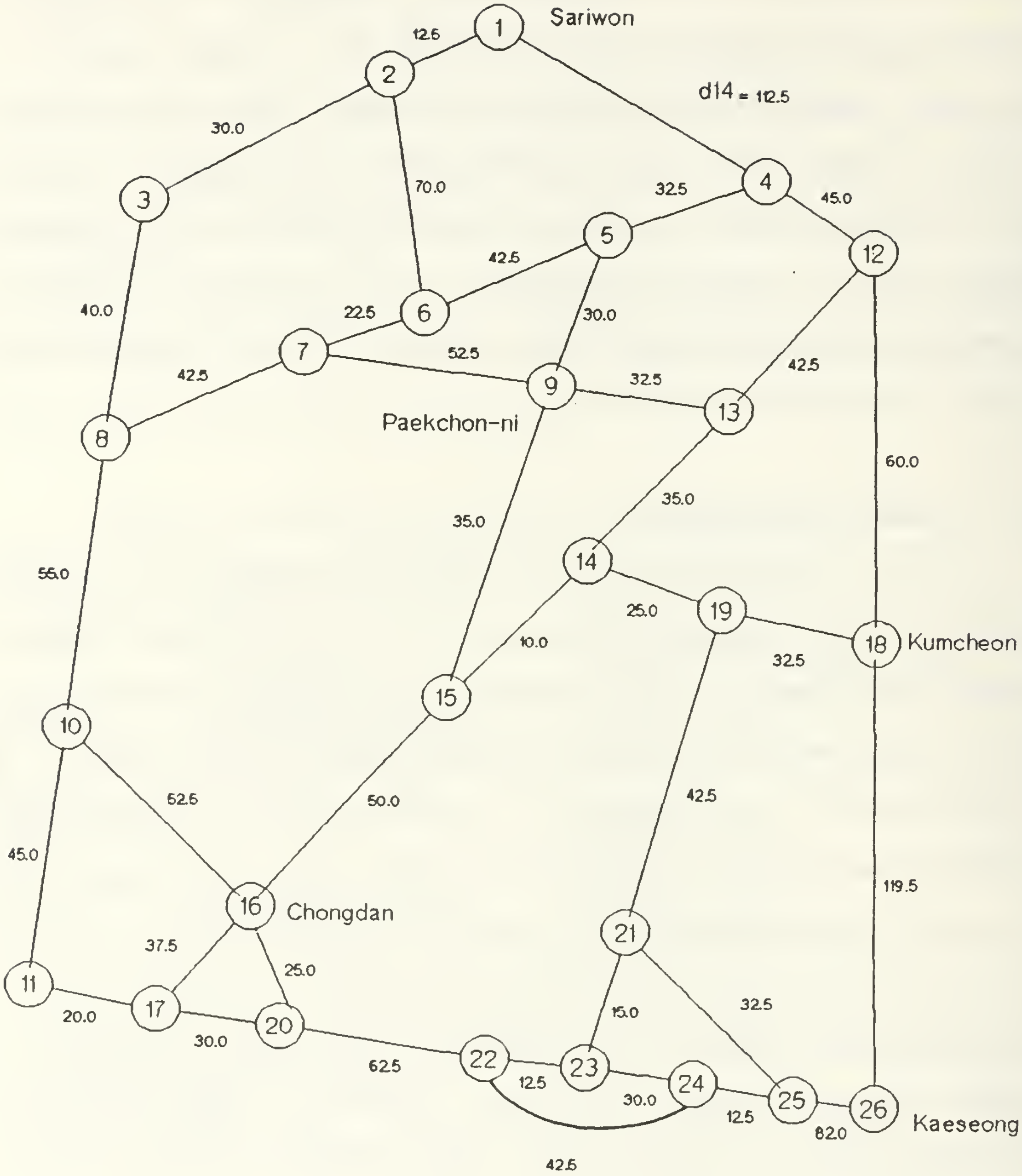

Figure 4. Example Network (Korea)

Data concerning North Korean equipment for transportation and road repair is unavailable in an unclassified form. In lieu of this data, we used the U. S. Army field manual [Ref. 8, 12, 13] to supply us with plausible data. We will now describe the development of the recovery and transit time data used in the network data file for the Korean scenario.

1. Recovery Time

The recovery time of an arc is the time required to repair a damaged road segment or to construct a bypass. We will assume that all network attacks are in the form of South Korean air strikes. We assume that eight sorties per day are available to attack the network and two sorties are allocated for each target. These are requirements which are imposed on strike aircraft operations in order to ensure safety of flight. Therefore, the number of arcs to be attacked is four.

Let's assume that there are five engineering units in this area: Sariwon (node 1), Packchon–ni (node 9), Chongdan (node 16), Kumcheon (node 18), and Kaeseong (node 26).

By using the assumptions stated above, the data in Table 4 of Appendix C was calculated. Column (1) of this table is the number of the arc (the nodes each arc connects to are shown in Columns 2 and 3 of Table 6). Column (2) represents the types of targets: bridges, mountain roads, tunnels, and roads. Column (3) is the distance in kilometers between the engineering unit and the target. Column (4) is the movement time calculated from column (3) and is stated in minutes. It is assumed that the speed of recovery personnel and equipment is 30 KM per hour. Columns (5) and (6) are repair time and bypass times expressed in minutes. Column (7) gives the minimum of the previous two columns. Column (8) is the recovery time which is the sum of columns (4) and (7). Column (9) gives the location of the nearest engineering unit to each target.

2. Transit Time and Cargo Requirements

As we discussed in Chapter III, we will choose open column formations for traversing all arcs. The density per KM is 12 trucks and the speed of each truck is 24 KM/H [Ref. 8: p.5–5]. The additional assumptions are no exogenous traffic and concrete or bituminous pavement.

Transit times are called road clearance times in military terminology. It is the total time a column or element thereof requires to travel over and clear either a section or all of a road. Road clearance time equals the time distance plus time length. Time distance is the time required to move from one point to another at a given rate of speed. Time length (pass time) is the time required for a column, or element thereof, to pass a given point [Ref. 8: p. F–2]. For application to the real network, the transit time over an arc is assumed to be only the time distance while the transit time of the last arc of each route, such as arcs 24 and 38, is the time length plus the time distance. This is due to the fact that the convoy column experiences the time length only at the end of their journey. Thus, only the last arc of each path needs to have time length added. The formula for time distance [Ref. 8: p. F–9] is:

$$\text{Time Distance} = \text{Distance} \;/\; \text{Rate} \;,$$

The formula for time length [Ref. 13: p. 3–57] is:

$$\text{Time Length} = (\text{number of vehicle} * 60) \;/\; (\text{density} * \text{rate}) + \text{time gaps} + \text{EXTAL},$$

where EXTAL is the extra time allowance.

In the above formulas, we know all of the required data except for the time gaps. For time gaps, we need the vehicle gaps. In the open column, the density is 12 trucks per KM and the length of a truck is 17 feet (1 feet is 0.3048 meters). Thus, the vehicle gap is:

$$\text{Vehicle Gap} = (1000-12 * 17 * 0.3048) / 11 = 85 \text{ (M)}.$$

In the open column, the traveling rate is 24 KM / H. So, the time gaps value is:

$$\text{Time Gaps} = (127 * 85 * 60) / 24000 = 27 \text{ (MIN)}.$$

The time length value is:

$$\text{Time Length} = (128 * 60) / (12 * 24) + 27 + 4 = 57 \text{ (MIN)}.$$

Table 5 includes the above information. Column (2) contains the length (Km) of each arc; Column (3) lists time distance; Column (4) lists time length; and Column (5) gives the transit time before interdiction.

3. Transit Time Comparison Between Two Models

Given the arc sets to be attacked, we will compare the results of two models. The deterministic model used expected values of recovery times and transit times, i.e.,

$$d(e) = 0.2d(1) + 0.2d(2) + 0.5d(3) + 0.1d(4),$$

$$c(e) = 0.2d(1) + 0.2d(2) + 0.5d(3) + 0.1d(4).$$

Outputs of the stochastic model was from 200 simulations. Table 7 of Appendix C shows the results of 10 arc sets selected as reasonable interdiction possibilities from the C_4^{38} choices(scenario 1). As we suggested in Chapter IV, the average network transit times from the stochastic model were less than those for the deterministic model except for the first, second, and tenth arc sets. The main reason for the reversal for these three arc sets is that the recovery times have less influence on the time–dependent transit time when the interdicted arcs are near the sink. So, in an attempt to increase the effects from such arcs, we multiplied recovery times by two and reran the simulation. The results are presented in table 8 (scenario 2) of Appendix C. As the reader can see in this table, the second and fourth arc sets still behave contrary to our expectations. The reason is that the arc sets include arcs 24 and 38 which are the last arcs of each route. Recovery times have no influence on the time–dependent transit time of these last arcs because they are repaired before the convoy reaches them.

C. EXAMPLE 2 (CALIFORNIA)

The assumptions of Example 2 are similar to those of North Korea. Therefore, we will use most of the same assumptions, and same procedures.

The area is between Gilroy and Carmel Valley Village. It is assumed that Carmel Valley Village is the frontal area and the logistics command is located at Gilroy. As we can see the Figure 5, the network is composed of 11 nodes and 15 arcs. The source node is Gilroy and the sink node is Carmel Valley Village.

1. Recovery Time

The structures which can be choke points are only bridges and roads in this area. It is assumed that six sorties per day are available in this area and two sorties are allocated for one target. Therefore, the number of arcs to be attacked is three in any one arc set.

Let's assume that there are three engineering units in this area: Gilroy (node 1), Castroville (node 6), and Carmel Valley Village (node 11). The data of Table 2 is assumed applicable in this area also. Table 9 of Appendix D shows recovery time data. The arc distances in Column (3) are expressed in miles. It is assumed that the speed of recovery personnel and equipment is 30 mile per hour (mph).

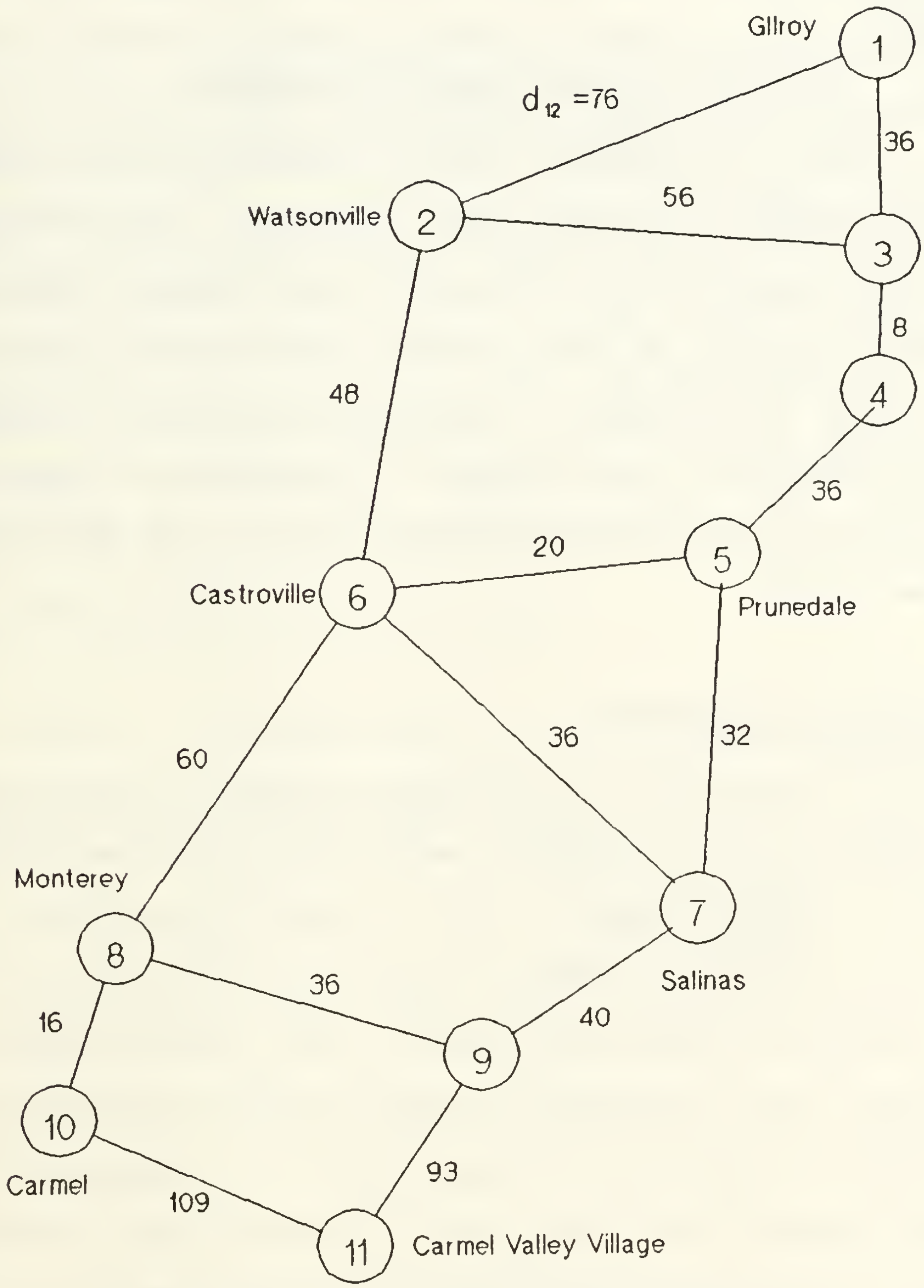

Figure 5. Example Network (California)

2. Transit Time

For the open column, the density per mile is assumed to be 20 and the speed is assumed to be 15 mph [Ref. 8: p.5–5]. Table 10 of Appendix D shows the transit time data. Table 11 of Appendix D is the network data file which applies for this example.

3. Transit Time Comparison Between Two Models

Table 12 of Appendix D shows of 10 arc sets selected as reasonable interdiction possibilities from the C_3^{15} choices (scenario 3). The average network transit times from the stochastic model were less than those for the deterministic model except for the first and ninth arc sets. The main reason is the same to scenario 1. Table 13 of Appendix D is for the case recovery time data is double that of scenario 3 (scenario 4). In case of scenario 4, every selected arc set has the same tendency, the transit times from the deterministic model is more than those from the stochastic model.

D. GENERAL COMMENTS

In all of the scenarios we see the general tendency is that the expected network transit time of the deterministic model is more than the average times of the stochastic model except when the arcs to be attacked are near the sink. This latter result is because the accumulated transit time to a node for such arcs is usually more than the recovery time of the arc. Only when the recovery times become large do those arcs recovery times affect the transit time through the net work as we saw in scenarios 2 and 4. We also notice that, in those cases where the deterministic expected time is larger than the average transit time from the simulations, the deterministic times are also larger than the upper bound of the 95

% confidence interval for the simulated times. Finally, we notice that the differences between transit times of two models is significant and the results are, at times, unpredictable.

The average decreases in transit time from the deterministic model to the stochastic model and the width of the confidence interval are shown in Table 14.

As we can see from these examples, the transit time calculated using the expected values is usually larger than the averages resulting from the stochastic simulation. Using expected values in place of random variables would lead the transportation manager to underestimate his ability to transport goods within a required delivery time. On the other hand, using the expected value model to determine the effectiveness of airstrikes to interdict flow of supplies leads to overestimating this effectiveness. In cases where this effective interdiction is critical, such as day four of a Korean conflict, this overestimation of effectiveness could cause serious losses of troops and territory.

TABLE 14. DECREASE RATE AND CONFIDENCE INTERVAL

Scenario	Avg. Decrease rate (%)	Range Of Width Of C.I (95 %)
1	7.33	(3.1625, 37.4108)
2	4.62	(3.1625, 17.8246)
3	13.13	(1.7914, 18.6705)
4	11.52	(1.7914, 36.6148)

VII. SUMMARY AND CONCLUSIONS

A. SUMMARY

The goal of military road network managers is to minimize the time required to transport war materials to their front line troops. The opposing or interdicting forces try to interdict some of the roads to maximize the transit time between the supply source and the front lines by using a variety of means, particularly airstrikes. If the transit time is greater than the required delivery time, the front line troops must either curtail their activity or retreat from the frontal area. Accordingly, it is to the advantage of the interdicting forces to delay the movement of materials as long as possible.

The assumption that any attack is considered completely successful is far from the real situation. The target damage may be complete, partial, or only ruts. Also aircraft may be shot down by anti–aircraft gunfire including defensive air patrol interceptors of the road managing forces. Probabilities of kill on the target vary with the levels of pilots' skill, the opponent's anti–air capability, the exposure time of aircraft, the type of munitions, hardness of targets, and so on. Therefore, the level of the success of an attack must be modeled as probabilistic instead of assuming that we always have completely successful destruction of the target.

If the relationships which comprise the model are simple enough, it may be possible to use analytical methods (such as algebra, calculus or probability theory) to obtain a direct optimal solution to questions of interest. However, most real–world systems are too complex to allow the models to be evaluated analytically, and the only resort left is to use computer simulation. In such a simulation we simulate a single occurrence of the events of the situation being modeled. Repeated simulations provide data which can be used to estimate the desired true characteristics of the model.

Our computer simulation program in Appendix B began by inputting the network data file which contained all of the relevant information for the problems of interest. We then entered a loop in which the pseudorandom attack successes were calculated. The resulting deterministic time–dependent shortest path problem was generated and solved using the Cooke and Halsey algorithm, and the resulting shortest path length was recorded. This loop was repeated 200 times for a given set of interdicted arcs. From the recorded statistics, the sample mean and sample variance of the resulting transit time was calculated and an output file was built which contained the results of the iterations, and the statistical summary.

B. CONCLUSIONS

The result of using the expected arc transit time values is to produce biased solutions. The transit time calculated using the expected values tends to be larger than the average values obtained from stochastic simulation. Using expected values in place of random variables would lead the transportation manager to underestimate his ability to transport goods within a required delivery time. On the other hand, using the expected value model to determine the effectiveness of airstrikes to interdict flow of supplies leads to overestimating this effectiveness when transit time over the shortest route is the main concern. In cases where this effective interdiction is critical, this overestimation of effectiveness could cause serious losses of troops and territory for either side. Therefore, the stochastic model should be used in the real situation.

APPENDIX A. RANDOM CAPACITY IN MAXIMUM FLOW PROBLEM

A. PROBLEM DESCRIPTION

We performed a simple experiment involving a small maximum flow problem on the network shown in Figure 3. We used coins and dice to generate random capacities and calculated the resulting random maximum flows.

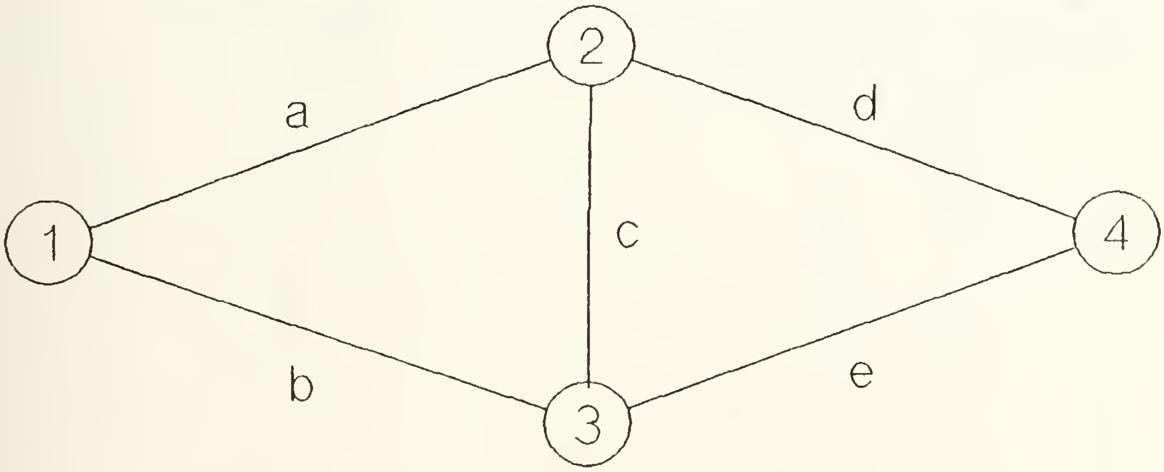

Figure 3. Maximum Flow Example Network

For random capacities, we used a dime, a quarter, a penny, a nickel, and a die simultaneously to generate the values of a, b, c, d, and e,respectively. If a coin flip results in a head, the capacity is 10, 25, 2, and 5 for arcs a, b, c, and d,respectively. If a result is a tail, the capacity is 1. As for arc e, the value is the face of die multiplied by two.

The expected value of the capacities are 5.5 for arc a, 13 for arc b, 1.5 for arc c, 3 for arc d, and 7 for arc e.

B. PROCEDURE AND OUTPUT COMPARISON

The maximum flow on the network using the expected capacities is 10. Table 3 shows the output of random maximum flows. z_1 is the single commodity maximum flow, and μ_i is the i sample mean after i independent experiments or trials. Thus,

$$\mu_i = \left(\sum_{j=1}^{i} z_{1j} \right) / i$$

TABLE 3. OUTPUT OF RANDOM MAXIMUM FLOWS

Trial	a	b	c	d	e	z_1	μ_i
1	10	25	1	5	2	7	7.00
2	1	1	2	5	6	2	4.50
3	10	1	1	1	2	3	4.00
4	10	25	2	5	8	13	6.25
5	10	25	1	5	2	7	6.40
6	1	25	1	1	8	9	6.83
7	1	25	2	1	10	11	7.43
8	1	1	1	1	12	2	6.75
9	10	25	1	1	4	5	6.56
10	1	1	1	1	12	2	6.10
11	1	25	1	1	4	5	6.00
12	1	25	1	5	4	5	5.92
13	1	1	2	5	8	2	5.62
14	10	1	2	1	2	3	5.43
15	1	25	2	1	8	9	5.66
16	1	25	2	1	6	7	5.75
17	1	25	1	5	4	5	5.71
18	10	25	2	1	10	11	6.00
19	1	1	1	5	10	2	5.79
20	10	25	1	1	8	9	5.95
21	10	25	1	1	12	13	6.29
22	10	25	1	1	8	9	6.41
23	1	25	2	5	4	9	6.52
24	1	1	1	1	4	2	6.33
25	1	1	1	1	8	2	6.16
26	10	25	2	5	4	9	6.27
27	10	25	2	1	6	7	6.30
28	1	25	1	5	4	5	6.25
29	1	25	1	1	4	5	6.21
30	10	1	2	5	6	8	6.27

As we see from the table, μ_i stabilizes somewhere near 6.25. The amount of flow calculated using the expected values is greater than the result of the thirty stochastic simulations by a factor of 37.3 per cent. Using expected values in place of random variables would lead the transportation manager to overestimate his ability to transport goods.

APPENDIX B. STOCHASTIC SIMULATION COMPUTER PROGRAM

```
      program thesis
c
c     Parameter statement must be identical in each subroutine except
c     for RAND  and STATS
      parameter(maxnod = 40, maxarc = 60, maxtyp = 5)
c
      parameter(kseed = 3645)
c
      real ctabl(maxnod, maxnod), dtabl(maxnod, maxnod),
     &     c(maxarc, maxtyp), d(maxarc, 0:maxtyp), cdf(maxarc, maxtyp),
     &     ztabl(maxnod, maxnod), sum, sumsq, label(maxnod)
c
      integer numnod, source, numtyp, numarc, atakd(maxarc),
     &     head(maxarc), tail(maxarc), dest, kk, iter
c
      character*20 fname
      character*1 z
c
      common last
c
c     Initialize the seed, sums, and the data file name
      last = kseed
      fname = 'netwrk2.dat'
      sum = 0.0
      sumsq = 0.0
      do 1019 i = 1, numnod
          do 1019 j = 1, numnod
              ztabl(i, j) = 0.0
1019  continue
c
c     Determine the number of simulation iterations to run
c
      Print *, 'How many iterations would you like, sir?'
      print *, 'Iterations = '
      read *, iter
      print *
      print *
c
c     Input the network structure in the subprogram READNET
c
      call READNET(fname, head, tail, c, d, numtyp, numnod,
     &    numarc, atakd, cdf, source, dest)
c
c     Debugging prints
c
c     print *
c     print *, 'Going from ', source, ' to ', dest
c     read (*,1)  z
      do 1011 k1 = 1, numarc
          if (atakd(k1).eq.1) then
              print *, 'Arc attacked: tail = ', tail(k1), ' head = ',
     &            head(k1), ' arc # ', k1
```

```
          endif
 1011  continue
 c
 c     9191 is the big loop for the iterations
 c
       do 9191 kk = 1, iter
 c
 c          Generate random construction and travel time tables
            call RTABLE(ctabl, dtabl, tail, head, c, d, atakd,
      &         cdf, numtyp, numnod, numarc)
 c
 c          Debugging print statements
 c          print *, 'Dtable and Ctabl from the main after RTABLE'
 c          print *
 c          do 1009 i = 1, numnod
 c              write(*, 10) (dtabl(i, j), j = 1, numnod)
 c009       continue
 c          write(*, *)
 c          do 1008 i = 1, numnod
 c              write(*, 10) (ctabl(i, j), j = 1, numnod)
 c008       continue
 c          read (*,1)  z
 1          format(a1)
 10         format(10(f8.3, 2x), / 10(f8.3, 2x) / 10(f8.3, 2x) /
      &             10(f8.3, 2x))
 c          print *, 'The cdf for ', numtyp, 'types of destruction'
 c          print *
 c          do 1010 i = 1, numarc
 c              print *, (cdf(i, j), j = 1, numtyp)
 c010       continue
 c
 c          Call the solver with the generated table
 c
            call SPSOLVE(numnod, dtabl, ctabl, source, label)
 c
 c          Debugging print statements
 c
 c          print *, 'destck ', dest, 'here come labels'
 c          do 23 k1 = 1, numnod
 c              print *, label(k1)
 c3         continue
 c          print *
 c
 c          Debugging print statements
 c          call outpt(label, numnod)
 c
 c          Collecting statistics on the performance
 c
            sum = sum + label(dest)
            sumsq = sumsq + (label(dest)**2)
 c          print *, 'Distance to source = ', label(dest)
 c          print *
```

```
C         print *
C         read (*, 1) z
9191  continue
C
C     Call the statistics subprogram
C
      call STATS(sum, sumsq, iter)
C
      stop
      end
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      subroutine STATS (sum, sumsq, iter)
C
      real sum, sumsq, riter
      integer iter
C
      riter = real(iter)
      print *, 'your statistics sir........'
      xbar = sum / riter
      print *, 'average = ', xbar
      if (iter.eq.1) then
          print *, 'thats all we get for one iteration '
          return
      endif
      stddev = sqrt((sumsq - ((sum ** 2) * (1/riter)))/(riter - 1))
      print *, 'standard deviation', stddev
      print *, 'stdev of x-bar = ', stddev / sqrt(riter)
      print *, '         for iter = ', iter
      rad1 = 1.96 * stddev / sqrt(riter)
      rad2 = 1.645 * stddev / sqrt(riter)
      print *, '95% confidence interval (',xbar - rad1,',',xbar
     &    + rad1,')'
      print *, '90% confidence interval (',xbar - rad2,',',xbar
     &    + rad2,')'
      print *
      return
      end
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      subroutine SPSOLVE(numnod, dtabl, ctabl, source, label)
C
      parameter(maxnod = 40, maxarc = 60, maxtyp = 5)
C
      integer numnod, source, curnod, node, prm(maxnod)
      real dtabl(maxnod, maxnod), ctabl(maxnod, maxnod),
     &    curtim, label(maxnod)
C
C     print *, 'Dtable and Ctable inside solver'
C     do 1009 i = 1, numnod
```

```
C          write(*, 10) (dtabl(i, j), j = 1, numnod)
C009   continue
C      write(*, *)
C      do 1008 i = 1, numnod
C          write(*, 10) (ctabl(i, j), j = 1, numnod)
C008   continue
C      read (*,1)  z
1      format(a1)
10         format(10(f8.3, 2x), / 10(f8.3, 2x) / 10(f8.3, 2x) /
     &              10(f8.3, 2x))
       do 1000 i = 1, numnod
           prm(i) = 0
           label(i) = 99999.0
1000   continue
C
C      print *, 'SOURCE = ', source
       prm(source) = 1
       label(source) = 0.0
       curnod = source
       curtim = 0.0
C
C   Find the time of travel from the current node to each of the
C   adjacent nodes which are not permanently labeled.  Find the minimum
C   distanced node, label it permanent.
C
       do 1001 iter = 1, numnod - 1
C        print *, 'iteration ', iter, 'current node ', curnod, 'time ',
C     &      curtim
C        print *, 'minlbl ', minlbl
         minlbl = 99999
C         read (*,1)  z
         do 1002 node = 1, numnod
             if (prm(node).eq.0) then
              if (curtim.ge.ctabl(curnod, node)) then
                 label(node)
     &       = min(label(node), curtim + dtabl(curnod, node))
              else
                 label(node)
     &       = min(label(node), ctabl(curnod,node) + dtabl(curnod,node))
              endif
              if (label(node).lt.minlbl) then
                 minlbl = label(node)
                 minnod = node
              endif
             endif
C
C
1002     continue
C
       curtim = label(minnod)
       prm(minnod) = 1
       curnod = minnod
```

```
c
c     do 9999 i = 1, numnod
c        print *, i, label(i)
c999  continue
c     read (*,1)  z
1001  continue
c     do 90 i = 1, numnod
c        print *, 'node ', i, 'prm(node) ', prm(i)
c0    continue
      return
      end
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      subroutine RTABLE(ctabl, dtabl, tail, head, c, d, atakd,
     &    cdf, numtyp, numnod, numarc)
c
      parameter(maxnod = 40, maxarc = 60, maxtyp = 5)
c
      real ctabl(maxnod, maxnod), dtabl(maxnod, maxnod),
     &   c(maxarc, maxtyp), d(maxarc, 0:maxtyp), cdf(maxarc, maxtyp),
     &   rand
      integer numnod, numarc, numtyp, tail(maxarc), head(maxarc),
     &    atakd(maxarc), type
      common last
c
c     Debugging print
c
c     print *, 'for arc 2: '
c     do 19 i = 1, numtyp
c         write(*,2) i,  c(2, i), d(2, i), cdf(2, i)
c9    continue
2     format(' destr type ', i3, ' c = ', f8.3,' d = ',f8.3,' cdf = ',
     &    f8.3)
c     read (*, 1) z
c
c     Initialize the table
c
      do 1001 i = 1, numnod
          do 1002 j = 1, numnod
              ctabl(i, j) = 99999
              dtabl(i, j) = 99999
1002  continue
1001  continue
c
c     Assign the proper construction and time-of-travels in the tables
c     ctabl and dtabl
c
      do 1005 i = 1, numarc
          if (atakd(i).eq.0) then
c             Assign the unharassed travel distance and no construct. time
              dtabl(head(i), tail(i)) = d(i, 0)
```

```
                ctabl(head(i), tail(i)) = 0.0
                dtabl(tail(i), head(i)) = d(i, 0)
                ctabl(tail(i), head(i)) = 0.0
            else
                p = rand()
C               Debug print
C
C               print *, '                  p = ', p
C
C               Determine the type of destruction from the random probability
C               p, and assign the construction and travel times accordingly
C
                do 1003 j = 1, numtyp
                    if (p.lt.cdf(i, j)) then
                        type = j
                        goto 1004
                    endif
1003            continue
1004            continue
                dtabl(head(i), tail(i)) = d(i, type)
                ctabl(head(i), tail(i)) = c(i, type)
                dtabl(tail(i), head(i)) = d(i, type)
                ctabl(tail(i), head(i)) = c(i, type)
            endif
1005    continue
C
C       Debug print
C
C       print *, 'Dtable and Ctable at the end of Rtable'
C       do 1009 i = 1, numnod
C           write(*, 10) (dtabl(i, j), j = 1, numnod)
C009    continue
C       write(*, *)
C       do 1008 i = 1, numnod
C           write(*, 10) (ctabl(i, j), j = 1, numnod)
C008    continue
C       read (*,1)  z
        return
1       format(a1)
10          format(10(f8.3, 2x), / 10(f8.3, 2x) / 10(f8.3, 2x) /
     &             10(f8.3, 2x))
        end
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
        subroutine READNET(fname, head, tail, c, d, numtyp, numnod,
     &     numarc, atakd, cdf, source, dest)
C
        parameter(maxnod = 40, maxarc = 60, maxtyp = 5)
C
        real c(maxarc, maxtyp), d(maxarc, 0:maxtyp), cdf(maxarc, maxtyp),
     &   prob(maxarc)
```

```
      integer numnod, numarc, numtyp, tail(maxarc), head(maxarc),
     &    atakd(maxarc), arc, type, source, dest
      character*20 fname
      character*80 chrlin
C
      open(unit = 20, file = fname)
C
C     Read the network data and the attack plan
C
      read(20, 100) chrlin
      read(20, *) source, dest
C     print *, 's = ', source, ' t = ', dest
      read(20, *) numnod, numarc, numtyp
      read(20, 100) chrlin
C     print *, numnod, numarc, numtyp
100   format(a80)
      do 1000 i = 1, numarc
          read(20, *) arc, tail(i), head(i), (c(i, type), type = 1,
     &          numtyp), (d(i, type), type = 0, numtyp), atakd(i)
1000  continue
C
C     Read the probability of damage of each type and compute the cdf
C
      read(20, 100) chrlin
C     print *, 'numarc = ', numarc
      do 1001 i = 1, numarc
          read(20, *) arc, (prob(type), type = 1, numtyp)
C          print *, i, arc, (prob(type), type = 1, numtyp)
          cdf(i, 1) = prob(1)
          do 1002 j = 2, numtyp
              cdf(i, j) = cdf(i, j - 1) + prob(j)
1002      continue
1001  continue
      return
      end
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      subroutine outpt(arrive, numnod)
C
      parameter(maxnod = 40, maxarc = 60, maxtyp = 5)
C
      real arrive(maxnod)
      print *
      print *, '========================================================'
      do 1000 i = 1, numnod
          print *, 'NODE ', i, '  time from source ', arrive(i)
1000  continue
      return
      end
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
c
        real function rand()
c
c       last is the last integer generated by t
      parameter (L = 29, C = 217, xm = 1024.0)
      real p
      common last
c
      last = mod(last * l + c, xm)
      rand = real(last)/xm
      return
      end
```

APPENDIX C. TABLES AND NETWORK DATA FILE (EXAMPLE 1)

TABLE 4. RECOVERY TIME DATA (KOREA)

Arc	Target	Dist	M.T.	Rep	Byp	Cons	Rec	E. L
1	road	2	4	300	240	240	244	node 1
2	road	20	40	300	240	240	280	node 1
3	bridge	10	20	300	.	300	320	node 1
4	bridge	20	40	300	.	300	340	node 1
5	mtn.road	30	60	360	420	360	420	node 1
6	mtn.road	18	36	360	420	360	396	node 9
7	road	34	68	300	240	240	308	node 9
8	mtn.road	20	40	360	420	360	400	node 9
9	mtn.road	24	48	360	420	360	408	node 9
10	bridge	30	60	300	.	300	360	node 9
11	mtn.road	6	12	360	420	360	372	node 9
12	tunnel	10	20	1200	.	1200	1220	node 9
13	mtn.road	21	42	360	420	360	402	node 9
14	road	6	12	300	240	240	252	node 9
15	mtn.road	7	14	360	420	360	374	node 9
16	bridge	33	66	300	.	300	366	node 16
17	bridge	8	16	300	.	300	316	node 18
18	road	25	50	300	240	240	290	node 9
19	road	16	32	300	240	240	272	node 9
20	bridge	4	8	300	.	300	308	node 18
21	mtn.road	18	36	360	420	360	396	node 18
22	bridge	10	20	300	.	300	320	node 16
23	mtn.road	10	20	360	420	360	380	node 16
24	bridge	12	24	300	.	300	324	node 18
25	mtn.road	22	44	360	420	360	404	node 18
26	road	5	10	300	240	240	250	node 16
27	road	8	16	300	240	240	256	node 16
28	mtn.road	30	60	360	420	360	420	node 16
29	road	19	38	300	240	240	278	node 16
30	road	16	32	300	240	240	272	node 16
31	bridge	8	16	300	.	300	316	node 26
32	road	13	26	300	240	240	266	node 26
33	bridge	15	30	300	.	300	330	node 16
34	road	24	48	300	240	240	288	node 26
35	road	30	60	300	240	240	300	node 26
36	road	21	42	300	240	240	282	node 26
37	bridge	12	24	300	.	300	324	node 26
38	road	5	10	300	240	240	250	node 26

TABLE 5. TRANSIT TIME DATA (KOREA)

Arc	Distance	Time Distance	Time Length	d(0)
1	5	13	.	13
2	45	113	.	113
3	12	30	.	30
4	28	35	.	35
5	16	40	.	40
6	13	33	.	33
7	18	45	.	45
8	17	43	.	43
9	9	23	.	23
10	17	43	.	43
11	12	30	.	30
12	21	53	.	53
13	17	43	.	43
14	13	33	.	33
15	14	35	.	35
16	22	55	.	55
17	24	60	.	60
18	14	35	.	35
19	4	10	.	10
20	13	33	.	33
21	10	25	.	25
22	20	50	.	50
23	21	53	.	53
24	25	63	57	120
25	17	43	.	43
26	10	25	.	25
27	15	38	.	38
28	18	45	.	45
29	8	20	.	20
30	12	30	.	30
31	13	33	.	33
32	6	15	.	15
33	25	63	.	63
34	17	43	.	43
35	5	13	.	13
36	12	30	.	30
37	5	43	.	43
38	10	25	57	82

TABLE 6. NETWORK DATA FILE

```
source, dest followed by numnod, numarc, numtyp
1  26
26 38 4
arc, T, H, c(1),c(2),c(3), c(4),  d(0), d(1),   .  .  , d(4), atk (0 = no)
1    1, 2,    0,  0, 146.4, 244,   12.5,  13.75, 17.5,  13.75,  14.375, 0
2    1, 4,    0,  0, 168.0, 280,  112.5, 123.75,157.5, 123.75, 129.375, 1
3    2, 3,    0,  0, 192.0, 320,   30.0,  33.00, 42.0,  33.00,  34.500, 1
4    2, 6,    0,  0, 204.0, 340,   70.0,  77.00, 98.0,  77.00,  80.500, 0
5    3, 8,    0,  0, 252.0, 420,   40.0,  44.00, 56.0,  44.00,  46.000, 0
6    4, 5,    0,  0, 237.6, 396,   32.5,  35.75, 45.5,  35.75,  37.375, 1
7    4, 12,   0,  0, 184.8, 308,   45.0,  49.50, 63.0,  49.50,  51.750, 0
8    5, 6,    0,  0, 240.0, 400,   42.5,  46.75, 59.5,  46.75,  48.875, 0
9    6, 7,    0,  0, 244.8, 408,   22.5,  24.75, 31.5,  24.75,  25.875, 0
10   7, 8,    0,  0, 216.0, 360,   42.5,  46.75, 59.5,  46.75,  48.875, 0
11   5, 9,    0,  0, 223.2, 372,   30.0,  33.00, 42.0,  33.00,  34.500, 0
12   7, 9,    0,  0, 732.0,1220,   52.5,  57.75, 73.5,  57.75,  60.375, 1
13  12, 13,   0,  0, 241.2, 402,   42.5,  46.75, 59.5,  46.75,  48.875, 0
14   9, 13,   0,  0, 151.2, 252,   32.5,  35.75, 45.5,  35.75,  37.375, 0
15   9, 15,   0,  0, 224.4, 374,   35.0,  38.50, 49.0,  38.50,  40.250, 0
16   8, 10,   0,  0, 219.6, 366,   55.0,  60.50, 77.0,  60.50,  63.250, 0
17  12, 18,   0,  0, 189.6, 316,   60.0,  66.00, 84.0,  66.00,  69.000, 0
18  13, 14,   0,  0, 174.0, 290,   35.0,  38.50, 49.0,  38.50,  40.250, 0
19  14, 15,   0,  0, 163.2, 272,   10.0,  11.00, 14.0,  11.00,  11.500, 0
20  18, 19,   0,  0, 184.8, 308,   32.5,  35.75, 45.5,  35.75,  37.375, 0
21  14, 19,   0,  0, 237.6, 396,   25.0,  27.50, 35.0,  27.50,  28.750, 0
22  15, 16,   0,  0, 192.0, 320,   50.0,  55.00, 70.0,  55.00,  57.500, 0
23  10, 16,   0,  0, 228.0, 380,   52.5,  57.75, 73.5,  57.75,  60.375, 0
24  18, 26,   0,  0, 194.4, 324,  119.5, 131.45,167.3, 131.45  137.425, 0
25  19, 21,   0,  0, 242.4, 404,   42.5,  46.75, 59.5,  46.75,  48.875, 0
26  16, 20,   0,  0, 150.0, 250,   25.0,  27.50, 35.0,  27.50,  28.750, 0
27  16, 17,   0,  0, 153.6, 256,   37.5,  41.25, 52.5,  41.25,  43.125, 0
28  10, 11,   0,  0, 252.0, 420,   45.0,  49.50, 63.0,  49.50,  51.750, 0
29  11, 17,   0,  0, 166.8, 278,   20.0,  22.00, 28.0,  22.00,  23.000, 0
30  17, 20,   0,  0, 163.2, 272,   30.0,  33.00, 42.0,  33.00,  34.500, 0
31  21, 25,   0,  0, 189.6, 316,   32.5,  35.75, 45.5,  35.75,  37.375, 0
32  21, 23,   0,  0, 159.6, 266,   15.0,  16.50, 21.0,  16.50,  17.250, 0
33  20, 22,   0,  0, 198.0, 330,   62.5,  68.75, 87.5,  68.75,  71.875, 0
34  22, 24,   0,  0, 172.8, 280,   42.5,  46.75, 59.5,  46.75,  48.875, 0
35  22, 23,   0,  0, 180.0, 300,   12.5,  13.75, 17.5,  13.75,  14.375, 0
36  23, 24,   0,  0, 169.2, 282,   30.0,  33.00, 42.0,  33.00,  34.500, 0
37  24, 25,   0,  0, 194.4, 324,   12.5,  13.75, 17.5,  13.75,  14.375, 0
38  25, 26,   0,  0, 150.0, 250,   82.0,  90.20,114.8,  90.20,  94.300, 0
arc,p(1),p(2),..., p(4)
1  0.2, 0.2, 0.5, 0.1
2  0.2, 0.2, 0.5, 0.1
3  0.2, 0.2, 0.5, 0.1
4  0.2, 0.2, 0.5, 0.1
5  0.2, 0.2, 0.5, 0.1
6  0.2, 0.2, 0.5, 0.1
7  0.2, 0.2, 0.5, 0.1
8  0.2, 0.2, 0.5, 0.1
9  0.2, 0.2, 0.5, 0.1
```

```
10 0.2, 0.2, 0.5, 0.1
11 0.2, 0.2, 0.5, 0.1
12 0.2, 0.2, 0.5, 0.1
13 0.2, 0.2, 0.5, 0.1
14 0.2, 0.2, 0.5, 0.1
15 0.2, 0.2, 0.5, 0.1
16 0.2, 0.2, 0.5, 0.1
17 0.2, 0.2, 0.5, 0.1
18 0.2, 0.2, 0.5, 0.1
19 0.2, 0.2, 0.5, 0.1
20 0.2, 0.2, 0.5, 0.1
21 0.2, 0.2, 0.5, 0.1
22 0.2, 0.2, 0.5, 0.1
23 0.2, 0.2, 0.5, 0.1
24 0.2, 0.2, 0.5, 0.1
25 0.2, 0.2, 0.5, 0.1
26 0.2, 0.2, 0.5, 0.1
27 0.2, 0.2, 0.5, 0.1
28 0.2, 0.2, 0.5, 0.1
29 0.2, 0.2, 0.5, 0.1
30 0.2, 0.2, 0.5, 0.1
31 0.2, 0.2, 0.5, 0.1
32 0.2, 0.2, 0.5, 0.1
33 0.2, 0.2, 0.5, 0.1
34 0.2, 0.2, 0.5, 0.1
35 0.2, 0.2, 0.5, 0.1
36 0.2, 0.2, 0.5, 0.1
37 0.2, 0.2, 0.5, 0.1
38 0.2, 0.2, 0.5, 0.1

* c(3) = 0.6 * c(4)
  d(1) = 1.1 * d(0)
  d(2) = 1.4 * d(0)
  d(3) = 1.1 * d(0)
  d(4) = 1.15 * d(0)
```

TABLE 7. TRANSIT TIME COMPARISON BETWEEN TWO MODELS
(Scenario 1)

Arc Set	Determin. Time	Stochastic		
		Time	C.I(95 %)	C.I(90 %)
4,12,16,17	346.9000	368.6438	(366.6758, 370.6118)	(366.9921, 370.2955)
24,25,26,28	356.7175	360.9398	(358.2446, 363.6351)	(358.6778, 363.2019)
2, 3, 6,12	377.0000	371.3937	(369.8125, 372.9750)	(370.0666, 372.7209)
2, 4,24,38	418.0300	401.7033	(399.0912, 404.3153)	(399.5110, 403.8955)
2, 4, 7,10	414.5000	401.2283	(399.0602, 403.3964)	(399.4086, 403.0479)
2, 4,10,38	427.0000	391.5468	(387.8984, 395.1952)	(388.4847, 394.6089)
2, 4,10,16	467.5625	404.2913	(397.2708, 411.3118)	(398.3990, 410.1835)
2, 4, 5,10	467.5625	416.2100	(407.2977, 425.1223)	(408.7300, 423.6899)
2, 4, 8,12	414.5000	393.2225	(389.7153, 396.7297)	(390.2789, 396.1661)
7,11,12,16	355.1250	379.3322	(375.0317, 383.6328)	(375.7228, 382.9416)

TABLE 8. TRANSIT TIME COMPARISON BETWEEN TWO MODELS
(Scenario 2)

Arc Set	Determin. Time	Stochastic		
		Time	C.I(95 %)	C.I(90 %)
4,12,16,17	397.0000	378.9950	(376.0747, 381.9153)	(376.5440, 381.4460)
24,25,26,28	398.4175	399.5087	(394.6451, 404.3723)	(395.4267, 403.5907)
2, 3, 6,12	377.0000	371.3937	(369.8125, 372.9750)	(370.0666, 372.7209
2, 4,24,38	418.0300	419.2902	(414.5211, 424.0593)	(415.2875, 423.2928)
2, 4, 7,10	414.5000	401.2563	(399.0849, 403.4276)	(399.4338, 403.0787)
2, 4,10,38	427.0000	391.5508	(387.9015, 395.2000)	(388.4880, 394.6136)
2, 4,10,16	579.0625	427.0912	(412.2838, 441.8987)	(414.6636, 439.5189)
2, 4, 5,10	579.5625	454.6700	(435.9646, 473.3754)	(438.9708, 470.3692)
2, 4, 8,12	414.5000	393.2225	(389.7153, 396.7297)	(390.2789, 396.1661)
7,11,12,16	478.3250	405.5908	(393.2462, 417.9355)	(395.2302, 415.9515)

APPENDIX D. TABLES AND NETWORK DATA FILE (EXAMPLE 2)

TABLE 9. RECOVERY TIME DATA (CALIFORNIA)

Arc	Target	Dist	M.T.	Rep	Byp	Cons	Rec	E. L
1	bridge	5	10	300	180	180	190	node 1
2	road	6	12	300	240	240	252	node 1
3	bridge	12	24	300	180	180	204	node 1
4	road	10	20	300	240	240	260	node 1
5	road	10	20	300	240	240	260	node 6
6	bridge	5	10	300	180	180	190	node 6
7	road	3	6	300	240	240	246	node 6
8	road	9	18	300	240	240	258	node 6
9	road	6	12	300	240	240	252	node 6
10	bridge	3	6	300	.	300	306	node 6
11	bridge	13	26	300	.	300	326	node 6
12	road	13	26	300	240	240	266	node 11
13	road	15	30	300	240	240	270	node 11
14	road	7	14	300	240	240	254	node 11
15	road	8	16	300	240	240	256	node 11

TABLE 10. TRANSIT TIME DATA (CALIFORNIA)

Arc	Distance	Time Distance	Time Length	d(0)
1	19	76	.	76
2	9	36	.	36
3	14	56	.	56
4	2	8	.	8
5	9	36	.	36
6	12	48	.	48
7	5	20	.	20
8	8	32	.	32
9	9	36	.	36
10	15	60	.	60
11	10	40	.	40
12	9	36	.	36
13	4	16	.	16
14	9	36	57	93
15	13	52	57	109

TABLE 11. NETWORK DATA FILE

```
source, dest followed by numnod, numarc, numtyp
1  11
11 15 4
arc, T, H,    c(1), c(2), ., c(4),  d(0), d(1), .., d(4), atk (0 = no)
1    1, 2,     0,   0, 114.0, 190,    76,   83.6, 106.4,  83.6,   87.40, 0
2    1, 3,     0,   0, 151.2, 252,    36,   39.6,  50.4,  39.6,   41.40, 0
3    2, 3,     0,   0, 122.4, 204,    56,   61.6,  78.4,  61.6,   64.40, 0
4    3, 4,     0,   0, 156.0, 260,     8,    8.8,  11.2,   8.8,    9.20, 0
5    4, 5,     0,   0, 156.0, 260,    36,   39.6,  50.4,  39.6,   41.40, 0
6    2, 6,     0,   0, 114.0, 190,    48,   52.8,  67.2,  52.8,   55.20, 0
7    5, 6,     0,   0, 147.6, 246,    20,   22.0,  28.0,  22.0,   23.00, 1
8    5, 7,     0,   0, 154.8, 258,    32,   35.2,  44.8,  35.2,   36.80, 1
9    6, 7,     0,   0, 151.2, 252,    36,   39.6,  50.4,  39.6,   41.40, 1
10   6, 8,     0,   0, 183.6, 306,    60,   66.0,  84.0,  66.0,   69.00, 0
11   7, 9,     0,   0, 195.6, 326,    40,   44.0,  56.0,  44.0,   46.00, 0
12   8, 9,     0,   0, 159.6, 266,    36,   39.6,  50.4,  39.6,   41.40, 0
13   8, 10,    0,   0, 162.0, 270,    16,   17.6,  22.4,  17.6,   18.40, 0
14   9, 11,    0,   0, 152.4, 254,    93,  102.3, 130.2, 102.3,  106.95, 0
15  10, 11,    0,   0, 153.6, 256,   109,  119.9, 152.6, 119.9,  125.35, 0
arc,p(1),p(2),..,p(4)
1  0.2, 0.2, 0.5, 0.1
2  0.2, 0.2, 0.5, 0.1
3  0.2, 0.2, 0.5, 0.1
4  0.2, 0.2, 0.5, 0.1
5  0.2, 0.2, 0.5, 0.1
6  0.2, 0.2, 0.5, 0.1
7  0.2, 0.2, 0.5, 0.1
8  0.2, 0.2, 0.5, 0.1
9  0.2, 0.2, 0.5, 0.1
10 0.2, 0.2, 0.5, 0.1
11 0.2, 0.2, 0.5, 0.1
12 0.2, 0.2, 0.5, 0.1
13 0.2, 0.2, 0.5, 0.1
14 0.2, 0.2, 0.5, 0.1
15 0.2, 0.2, 0.5, 0.1

* c(3) = 0.6 * c(4)
  d(1) = 1.1 * d(0)
  d(2) = 1.4 * d(0)
  d(3) = 1.1 * d(0)
  d(40 = 1.15 * d(0)
```

TABLE 12. TRANSIT TIME COMPARISON BETWEEN TWO MODELS
(Scenario 3)

Arc Set	Determin.	Stochastic		
	Time	Time	C.I(95 %)	C.I(90 %)
8, 9, 10	273.4800	284.2860	(280.3159, 288.2562)	(280.9539, 287.6182)
1, 2, 6	351.7400	335.0332	(325.6980, 344.3685)	(327.1983, 342.8682)
1, 2, 4	353.0600	344.0573	(335.7219, 352.3926)	(337.0615, 351.0530)
1, 2, 14	367.0850	346.1183	(337.0435, 355.1932)	(338.5019, 353.7347)
1, 2, 5	357.6800	346.9688	(338.6827, 355.2550)	(340.0144, 353.9233)
2, 3, 7	309.0000	279.2960	(276.6637, 281.9283)	(277.0867, 281.5053)
1, 2, 8	357.0200	338.7611	(329.9167, 347.6056)	(331.3381, 346.1841)
1, 2, 11	358.3400	346.1733	(337.6795, 354.6671)	(339.0446, 353.3020)
7, 8, 9	273.4800	282.9700	(279.5856, 286.3544)	(280.1295, 285.8105)
2, 5, 8	293.0000	291.0300	(290.1343, 291.9257)	(290.2783, 291.7817)

TABLE 13. TRANSIT TIME COMPARISON BETWEEN TWO MODELS (Scenario 4)

Arc Set	Determin. Time	Stochastic		
		Time	C.I(95 %)	C.I(90 %)
8, 9, 10	376.5400	323.8200	(311.1291, 336.5109)	(313.1687, 334.4713)
1, 2, 6	452.5400	397.9328	(379.6254, 416.2402)	(382.5677, 413.2979)
1, 2, 4	453.8600	401.3493	(384.2001, 418.4986)	(386.9562, 415.7424)
1, 2, 14	467.8850	409.0539	(391.7082, 426.3995)	(394.4959, 423.6118)
1, 2, 5	457.5400	403.1413	(386.1323, 420.1503)	(388.8659, 417.4167)
2, 3, 7	293.0000	279.2960	(276.6637, 281.9283)	(277.0867, 281.5053)
1, 2, 8	457.5400	395.0933	(377.4376, 412.7490)	(380.2751, 409.9115)
1, 2, 11	459.1400	404.1253	(386.7449, 421.5057)	(389.5382, 418.7124)
7, 8, 9	309.0000	282.9700	(279.5856, 286.3544)	(280.1295, 285.8105)
2, 5, 8	293.0000	291.0300	(290.1343, 291.9257)	(290.2783, 291.7817)

LIST OF REFERENCES

1. Nugent, R. O., *Optimal Allocation of Air Strikes against a Transportation Network for an Exponential Damage Function*, Master's Thesis, Naval Postgraduate School, Monterey, California, June 1969.

2. Sullivan, M. E., *Maximizing Transit Times by Interdiction of a Transportation Network*, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1971.

3. Cooke, K. L. and Halsey, E., "The shortest Route through a Network with Time–Dependent Internodal Transit Times," *J Math. Anal. and Appl.* v. 14, pp. 493–498, 1966.

4. Mustin, T. M., *Optimal Allocation of Air Strikes for Interdiction of a Transportation Network*, Master's Thesis, Naval Postgraduate School, Monterey, California, June 1967.

5. Wollmer, R. D., "Algorithms for Targeting Strikes in a Lines–of–Communications Network," *Operations Research*, v. 18, no. 3, May–June 1970

6. Sengoku, M., Shinoda, S., and Yatsuboshi, R., "On a Function for the Vulnerability of a Directed Flow Network," *Networks*, v. 18, pp. 73–83, 1988.

7. Barrett, R., and others, "A Functional Overview of the Network Interdiction Model (NIM) Decision Support System (DSS)," *Working paper*, Center for Naval Analysis, Alexandria, Virginia, 1984.

8. *Army Motor Transport Units and Operations*, Headquarters, Department of the Army, FM 55–30, March 1980.

9. Law, A. M., and Kelton, W. D., *Simulation Modeling and Analysis*, McGraw–Hill, Inc., 1982.

10. Wagener, J. L., *FORTRAN 77 Principles of Programming*, John Wiley & Sons, Inc., 1980

11. Press, W. H., and others,*Numerical Recipes*,pp. 191–226, Cambridge University Press, 1986.

12. *Organizational, Technical, and Logistical Data Planning Factors*, Headquqrters, Department of the Army, Fm 101–10–1 / 2, October 1987.

13. *Transportation Reference Data*, Headquarters Department of the Army, FM 55–15, June 1986.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Commandant U.S. Army Transportation School Fort Eustis, Virginia 23604-5399	1
4.	Prof. Michael P. Bailey, Code 55Ba Department of Operations Research Naval Postgraduate School Monterey, California 93943-5004	1
5.	Prof. Alan W. McMasters Code 54Mg Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5004	1
6.	Prof. Samuel H. Parry, Code 55Py Department of Operations Research Naval Postgraduate School Monterey, California 93943-5004	1
7.	CPT Timothy P. Gannon P.O. Box 465 Martin, South Dakota 57551	1
8.	Defense Logistics Studies Information Exchange U.S. Army Logistics Management Center Fort Lee, Virginia 23801	1
9.	Commander TRADOC Analysis Command ATTN: ATRC-FA Fort Leavenworth, Kansas 66027-5200	1

10. Choi, Byung Kwon 10
152–030
Seoul–Si, Guro–Gu, Si–Heung 1 Dong,
HanYang Apt. 5–1107
Seoul, Korea

11. CPT. Kim, Suk Hwan 1
SMC 2318
Naval Postgraduate School
Monterey, California 93943–5000

12. Library 1
130–02
Korea Military Academy
Gong–Reung Dong, 556–21
Seoul, Korea

Thesis
C448843 Choi
c.1
Why stochastic modeling is essential in analyzing interdicted transportation network performance.

13 NOV 90 80247
13 NOV 90 80247
4 MAY 92 37541

Thesis
C448843 Choi
c.1
Why stochastic modeling is essential in analyzing interdicted transportation network performance.

thesC448843
Why stochastic modeling is essential in
3 2768 000 82965 9
DUDLEY KNOX LIBRARY